High Level Design (HLD)

Blog Creator Web Application

– BlogVerse –

Revision Number: 2.0
Last date of revision: 22/09/2023

Document Version Control

Date Issued	Version	Description	Author
26/08/2023	1	Initial HLD - V1.0	Abhijit Paul
22/09/2023	2	Final HLD - V2.0	Abhijit Paul

Contents

Abstract

In an increasingly interconnected world, online platforms have become pivotal for personal expression and community engagement.

This abstract offers an overview of a dynamic blogging website designed to facilitate user interaction through login, post creation, post viewing, and commenting functionalities.

Our blogging website prioritises user experience, offering a seamless login process that ensures the security and privacy of every user.
Upon login, individuals gain access to a versatile platform that empowers them to create and share their thoughts, experiences, and ideas with the global community.

The commenting system is a cornerstone of the platform, fostering discussions and connections among users. With an emphasis on respectful and constructive discourse, users can express their thoughts, offer feedback, and build relationships within the blogging community.

In summary, our user-centric blogging website offers a holistic and user-friendly experience, empowering individuals to login, create, explore, and engage. It serves as a digital haven for personal expression, knowledge sharing, and community building, connecting people worldwide through the power of words and ideas.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradiction prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design expects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include designs which are the architecture of the project
- List and describe the non functional attributes like:
    - ☐ Security
    - ☐ Reliability
    - ☐ Maintainability
    - ☐ Portability
    - ☐ Reusability

1.2 Scope

The HLD documentation presents structure of the system, such as database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mid-technical terms which should be understandable to the administrator of the system.

1.3 Definitions

HTML - HTML is the standard markup language for creating Web pages
CSS - CSS is a style sheet language used for describing the presentation of HTML
Bootstrap - Bootstrap is a free and open-source CSS framework
Jinja - Jinja is a web template engine for the Python programming language.

2 General Description

2.1 Product Perspective

The Blogging Website is a web application developed using Python Flask, designed to provide a platform for users to create, publish, and interact with blog posts.

2.2 Problem Statement

To create a web application to facilitate the blog creation capabilities.

- Create infrastructure in such a way that any user can create their blog.
- Allow users to use different types of formatting for text.
- Your web application should allow user to generate a link for every blog created at your application so that users can showcase their blog to the real world.

2.3 Proposed Solution

The solution proposed here is a web application based on Flask technology supporting the backend framework and HTML, CSS, JINJA templating and constructing the front end architechture. Users can view a gist of blogs posted on the platform. But to view the entire blog, users have to register on the platform which will ensure user engagement. Email verified users will have access to login to the platform and create blogs using HTML formatting tags, update blogs and view other's blogs. Users will be able to share the link of their blogs to others.

2.4 Further Improvements

Blogging platform can be added with more use cases such as Blog Likes, Adding formatting tools to format blog contents, User Profile section, Forgot and Update Password, Google Sign In.

2.5 Tools used

Python programming language and libraries such as Flask, FlaskLogin, BluePrints, Werkzeug, PyMongo, HTML, CSS, BootStrap are used to build the whole web application.

- Visual Studio Code is used as an IDE.
- For Front End, HTML, CSS, BootStrapping are used.
- For Backend Server, Flask framework is used.
- SQLite database is used to insert, retrieve, update and delete Blogs & User data.
- GitHub is used as a version control system.

2.6 Constraints

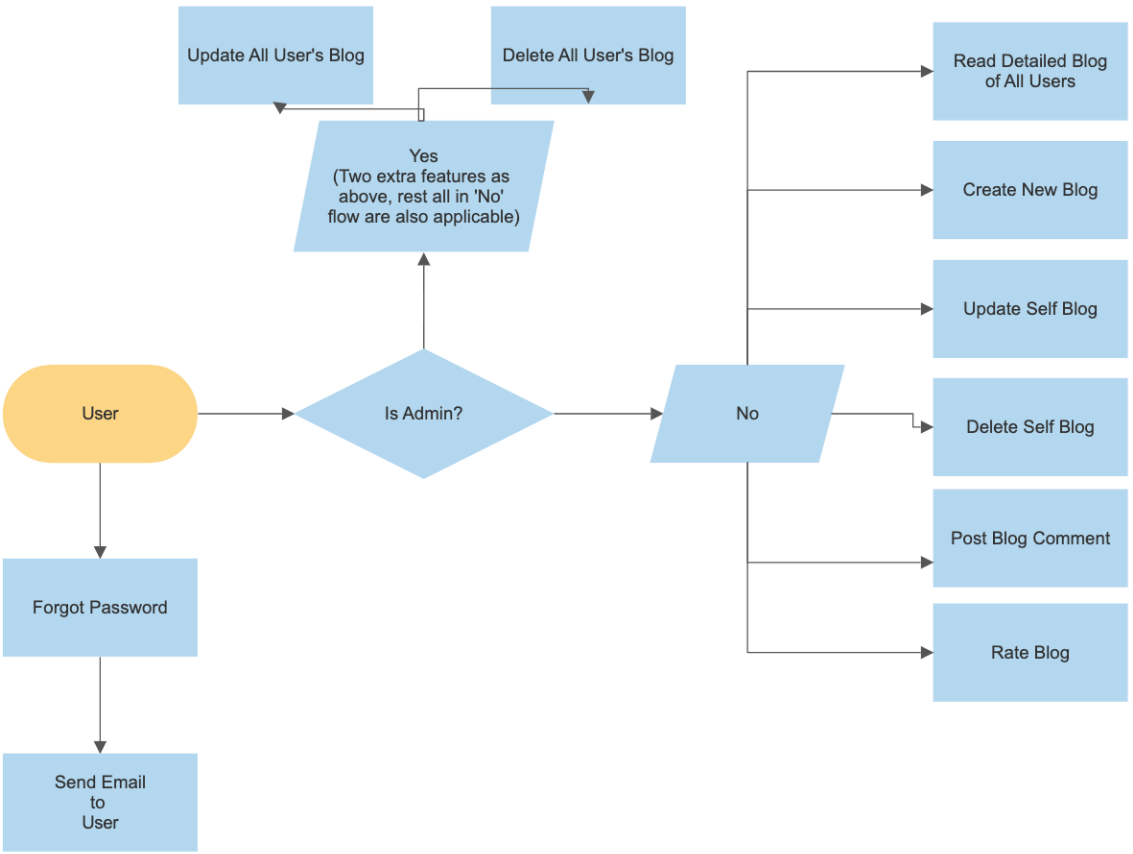
This project uses SQLAlchemy which is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL, connected with SQLite database which is native in nature, hence the web application could not be deployed in the cloud.

2.7 Assumptions

The main objective of this project is to implement the use-cases as previously mentioned (2.2 Problem Statement) to facilitate the blog creation capabilities.The front end web interface is created using HTML, CSS & BootStrap. Users are allowed to use all HTML tags to format the contents of the blog posts.

3 Design Details

3.1 Process Flow



3.2 Event log

3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong. An error will be defined as anything that falls outside the normal and intended usage.

4 Performance

4.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

4.2 Application Compatibility

The different components of the Blogging application will be using Python as an interface between them. All the components will have their own task to perform and it's the job of the Flask server to parse the instructions given by the client or user.

4.3 Resource Utilisation

When any task is performed, it will likely use all the required processing power until that task is finished.

4.4 Deployment

5 Conclusion

The blogging website will serve CRUD(Create, Read, Update and Delete) functionality by using the Flask web framework in Python. It will prioritise user experience, offering a seamless login process that ensures the security and privacy of every user, so that users can have a pleasant environment and enjoy blogging!